

# Содержание

- feature branch workflow** ..... 3
- Пошаговый процесс (Step-by-Step)** ..... 3
- 1. Создание задачи (Issue) ..... 3
- 2. Создание локальной ветки ..... 3
- 3. Первый коммит и создание Pull Request (PR) ..... 3
- 4. Обсуждение и разработка ..... 4
- 5. Ревью и слияние (Merge) ..... 4



# feature branch workflow

**Feature Branch Workflow** - это значит, что main/master всегда стабилен, а любая новая задача (фича / фикс / рефакторинг / и тд) разрабатывается в отдельной изолированной ветке.

Вот [тут](#) достаточно хорошо описано.

## Пошаговый процесс (Step-by-Step)

### 1. Создание задачи (Issue)

1. Создать Issue с названием задачи (если такой еще нет)
2. Хотя бы кратко, но лучше подробнее, описать задачу (можно копировать с PM если там оно полное)
3. Назначить себя на Issue (Assignees)
4. Issue получит уникальный номер в этом репозитории (#N напротив названия), можно запомнить, можно не запоминать (при написании # в описаниях/комментариях должен работать auto-suggestion)

### 2. Создание локальной ветки

Перед началом создается локальная ветка на основе актуального состояния main. Название ветки дается в формате: <FeatureSubject>[-optional-feature-fescription] (например Packaging-0.0.1, Packaging-fix-something).

```
git checkout main
git pull origin main
git checkout -b <feature branch>
```

### 3. Первый коммит и создание Pull Request (PR)

В нашей команде мы создаем PR как можно раньше, часто сразу после первого осмысленного коммита. (чтобы: сразу было видно над чем ведется задача, сразу можно было начать обсуждение по реализации и ссылаться на код из PR, в случае чего заметить что что-то не так с выбранным способом реализации как можно раньше)

Сообщение для коммита дается в формате: <CommitSubject> | <required-short-description> (например Packaging | add deb package build script, просто add <filename> лучше не делать, эта информация и так доступна внутри коммита, и новой осмысленной информации не несет)

Сделать первый коммит:

```
<code bash>
git add .
```

```
git commit -m "<commit message>"
git push -u origin <feature branch>
</code>
```

Создать PR (в названии можно просто повторить название issue).

В описании достаточно написать `Closes #<issue number>` (например `Closes #1`).

Если PR в `main` все остальное Gitea должен сделать сам. А именно: в Issue вместо `No Branch/Tag Specified` должно теперь быть название PR. Если нет, то лучше поставить вручную.

Назначить себя на PR (Assignees) и можно сразу назначить ревьюера (Reviewers).

Опционально, можно пометить PR как Draft (явно дать понять, что не готов к merge в `main`). Видимо на gitea это делается добавлением префикса WIP: к названию PR.

## 4. Обсуждение и разработка

Теперь:

1. В **Issue** обсуждаем \*бизнес-логику\* и общие вопросы (что сделать? как можно сделать?).
2. В **PR** обсуждаем \*реализацию\* (как сделано?).

## 5. Ревью и слияние (Merge)

Когда задача готова:

1. Снять статус Draft.
2. Запросить ревью (поискать кнопку).
3. Если после ревью были запрошены изменения, обсудить, внести и запустить эти изменения, запросить ревью опять. При необходимости повторить.
4. Когда замечаний больше нет и все доработки запущены слить в main (кнопка Create merge commit).

From:

<https://wiki.radi0.cc/> - radi0wiki

Permanent link:

[https://wiki.radi0.cc/glossary:prog:feature\\_branch\\_workflow](https://wiki.radi0.cc/glossary:prog:feature_branch_workflow)

Last update: **2026/04/09 17:07**

